

УДК 681.3

**Гимадеев К.Д., Теплинский С.В., Франжев К.Е.**  
Донецкий национальный технический университет  
Кафедра компьютерной инженерии  
E-mail:[kiria-b2@mail.ru](mailto:kiria-b2@mail.ru)  
[disep@mail.ru](mailto:disep@mail.ru)

## АНАЛИЗ И ИССЛЕДОВАНИЕ ТАЙМЕРОВ В WINDOWS

*Гимадеев К.Д., Теплинский С.В., Франжев К.Е. Анализ и исследование таймеров в Windows и C#. В статье рассматривается вопрос использования таймеров операционной системы Windows в языке программирования C#. Исследованы потери сообщений о тике таймера и сделаны предположения о некоторых аспектах работы таймера. Также исследована работа мультимедиа таймера.*

### **Введение**

Таймер в Windows является устройством ввода информации, которое периодически извещает приложение о том, что истек заданный интервал времени. В системе Windows таймер посылает сообщение WM\_TIMER, которое можно обработать в программе, тем самым выполняя заданные действия с периодичностью, указанной при инициализации таймера. Интервал таймера может находиться в пределах (теоретически) от 1 до 4294967295 миллисекунд, что составляет около 50 дней [2].

В библиотеке .NET Framework представлены 3 класса с таймерами в различных пространствах имён. Это System.Timers.Timer, System.Threading.Timer, System.Windows.Forms.Timer [3]. Данные классы реализуют системные таймеры, которые через заданный промежуток времени посылают системе синхронные сообщения, с приходом которых вызывается обработчик таймера.

### **Работа таймеров**

Все 3 класса устроены одинаково, но предназначены для разных нужд:

```

Код C#:
using System.Threading;
Timer timer1 = new Timer(delegate{ tics++; }, null, 0, 500);
Timer timer2 = new Timer(delegate{
Console.WriteLine(tics.ToString()); tics = 0;},null, 0, 10000);

```

- `System.Threading.Timer` – самый простой класс таймера, используется для выполнения команд через заданный промежуток времени.

- `System.Timers.Timer` – таймер оптимизированный для работы в многопоточных приложениях или приложениях, имеющих асинхронные функции

- `System.Windows.Forms.Timer` – простой таймер, работающий в платформе Windows Forms.

Пример инициализации таймера `System.Threading.Timer` в листинге 1.

Листинг 1 – инициализация таймера.

Данная программа выводит в консоль через каждые 10 секунд количество пришедших сообщений таймера `timer1`. На интервале в 10000 мс, который отсчитывается таймером `timer2`, имеем такие данные о количестве тиков таймера `timer1`, приведённые в таблицах 1,2,3.

Таблица 1 – данные о потерях сообщений таймера `timer1` класса `System.Threading.Timer` на интервале 10с (10000мс)

Интервал таймера <code>timer1</code>	500мс	100мс	50мс	10мс	1мс
Количество полученных (ожидаемых) сообщений	20(20)	92(100)	160(200)	637(1000)	640(10000)
Потери сообщений в процентах	0%	8%	20%	36%	93,6%

Таблица 2 – данные о потерях сообщений таймера timer1 класса System.Timers.Timer на интервале 10с (10000мс)

Интервал таймера timer1	500мс	100мс	50мс	10мс	1мс
Количество полученных (ожидаемых) сообщений	19 (20)	91(100)	160(200)	637(1000)	639(10000)
Потери сообщений в процентах	5%	9%	20%	36%	93,6%

Таблица 3 – данные о потерях сообщений таймера timer1 класса System.Windows.Forms.Timer на интервале 10с (10000мс)

Интервал таймера timer1	500мс	100мс	50мс	10мс	1мс
Количество полученных (ожидаемых) сообщений	20(20)	92(100)	160(200)	637(1000)	640(10000)
Потери сообщений в процентах	0%	8%	20%	36%	93,6%

Проанализировав данные таймеры, мы пришли к выводу, что системные таймеры в различных пространствах имен C# работают одинаково. Более того, эксперименты с таймером WinAPI дали такие же результаты.

В литературе [1,2] приводятся данные о том, что сообщения системного таймера приходят с частотой 18,21 раза в секунду, т.е. минимальный интервал составляет 55мс. Наши эксперименты показывают, что интервал таймера можно устанавливать с дискретностью 1мс. Однако, как видно из таблиц, интервал менее

10 мс не влияет на количество полученных сообщений WM\_TIMER.

Можно предположить, что такая потеря сообщений таймера может быть вызвана задержками на обработку сообщения таймера или другими причинами. Если в очереди сообщений появляются два и более сообщений от одного таймера, то они объединяются в одно. Вероятнее всего, другой причиной является регенерация памяти, которая запускается с интервалом не более 64мс [5]. В некоторых источниках указано, что частота интервала меньше 10 мс может отрицательно сказываться на работоспособности системы, поэтому в реализации таймера может стоять ограничение в 10мс.

### Мультимедиа таймер

Для повышения быстродействия нами был использован мультимедиа таймер из библиотеки функций WinAPI. Нами были проведены тесты, которые показывают, что таймер, работающий на частоте 1кГц, обеспечивает достаточную скорость работы и минимальную потерю сообщений WM\_TIMER при работе.

Таблица 4 - данные о потерях сообщений мультимедийного таймера

Интервал таймера	500мс	100мс	50мс	10мс	1мс
Количество полученных (ожидаемых) сообщений	20 (20)	100 (100)	200(200)	1000(1000)	10000 (10000)
Потери сообщений в процентах	0%	0%	0%	0%	0%

В стандартных средствах библиотеки .NET Framework нет мультимедиа таймера, поэтому мы импортировали некоторые функции из стандартной библиотеки WinAPI как показано в листинге 2.

Листинг 2 – импорт библиотек для подключения мультимедиа таймеа в C#

Код C#:

```
[DllImport("winmm.dll")]
private static extern int timeSetEvent(int delay, int resolution,
TimerEventHandler handler, IntPtr user, int eventType);
[DllImport("winmm.dll")]
private static extern int timeKillEvent(int id);
[DllImport("winmm.dll")]
private static extern int timeBeginPeriod(int msec);
[DllImport("winmm.dll")]
private static extern int timeEndPeriod(int msec);
```

При работе таймера, длительностью 10000 мс он посылает 10000 сообщений Windows которые успешно обрабатываются и не теряются с системной очереди сообщений.

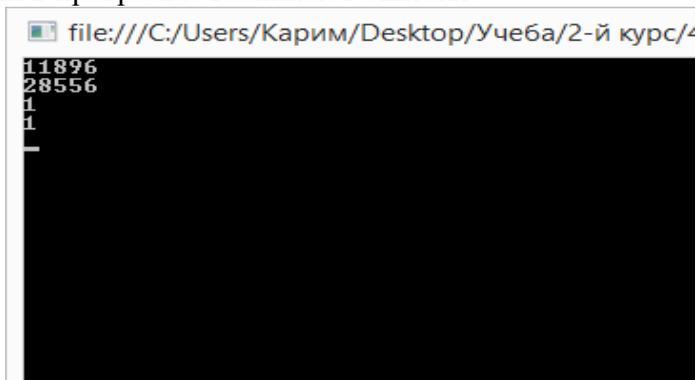
В основе всех системных таймеров лежит аппаратный программируемый таймер, максимальная частота которого от 1МГц и выше. Он управляет системными прерываниями и сообщениями WM\_PAINT. Можно предположить, что мультимедиа таймер отправляет асинхронные сообщения, которые не становятся в очередь и обрабатываются моментально, что позволяет достичь такой высокой частоты работы приложения. Однако это может сказываться на производительности системы в целом, что в свою очередь на современных компьютерах практически не заметно.

Для большинства программ, в которых используется таймер, такая высокая частота работы таймера не требуется. Однако такая частота может потребоваться при написании специализированных программ (работа с аудио/видео, обновление окна для плавной анимации). В таких случаях программист должен сам выбирать между обычным таймером и мультимедиа таймером, в зависимости от требуемой частоты работы таймера.

Необходимо учитывать тот факт, что при выполнении операций, на выполнение которых затрачивается времени больше, чем интервал тика таймера, могут случаться сбои в работе

программы. Это может быть связано с потерей сообщений мультимедиа таймера. Во избежание сбоев, программист должен учитывать время, необходимое на выполнение операций, исполняемых по тикку таймера.

Как видно на рисунке 1, при выполнении ресурсоёмкой операции в программе возникают ошибки.



```
file:///C:/Users/Карим/Desktop/Учеба/2-й курс/
11896
28556
1
1
-
```

Рисунок 1 – работа таймера с ресурсоёмкой операцией.

### Выводы

В ходе работы были исследованы различные системные таймеры, предоставленные в библиотеке .NET Framework, а так же мультимедиа таймер. Написаны программы для исследования таймеров в консоли и проекте WinForm.

Рассмотрен вариант подключения мультимедиа таймера для операций, требующих высокой частоты работы таймера.

### Список литературы

[1] Финогенов К.Г. Win32. Основы программирования. – 2-е изд., испр. и дополн. – М.: ДИАЛОГ – МИФИ, 2006. – 416с.

[2] Петзолд Ч. Программирование для Windows® 95 в двух томах

[3] Ресурс MSDN. / Интернет-ресурс. - Режим доступа : URL: <http://msdn.microsoft.com>

[4] Форум начинающих и профессиональных программистов / Интернет-ресурс. - Режим доступа : URL: <http://cyberforum.ru>

[5] Настройка подсистемы памяти в BIOS SETUP / Интернет-ресурс. - Режим доступа : URL: <http://www.ixbt.com/mainboard/memorytune-viabios.shtml>